

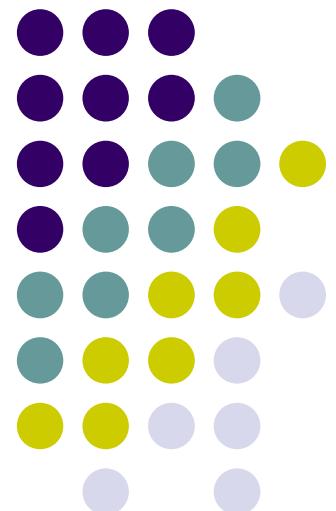
# **WIX1002**

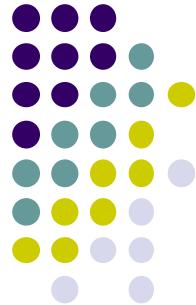
# **Fundamentals of Programming**

---

## **Chapter 6**

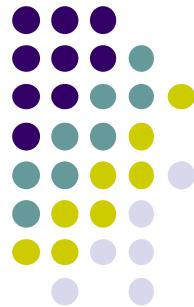
## **Java Methods**





# Contents

- Methods
- Static Method
- Reference Type



# Methods

- Modules in Java are called methods and classes. Methods allow the programmer to modularize a program.
- To define the method

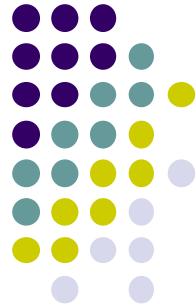
```
accessSpecifier returnType
```

```
  methodName(parameterType parameterName, ...) {
```

```
    // Method body
```

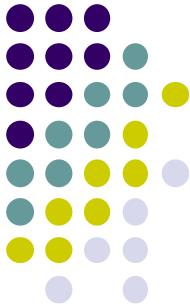
```
}
```

- The accessSpecifier for methods is usually public.
- The returnType specified the type of variable return by the method. It can be the **primitive type or object type**. Use **void** if the method return nothing.



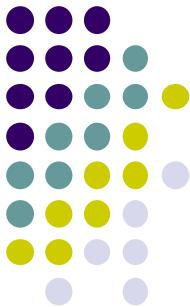
# Methods

- The body of a method that returns a value must contain at least one **return statement**.
- A return statement always end a method invocation.
- To return a value
  - **return variableName;**
- To end a void method, the return statement without any expression can be used
  - **return;**



# Methods

- Most methods have a list of parameters. The parameters provides the means for communicating information between methods via method calls.
- Methods can consists of one or more parameters or without parameter.
- A method is invoked by a method call. The method call specifies the **name of the method** and provides the correct parameters according to **parameter sequence and type**.
- When the method call completes, the method either returns a result to the calling method or simply returns control to the calling method.



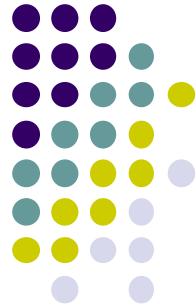
# Methods

- A variable declared within a method is called a **local variable**.
- Each method can have the same variable name, however, these variables are different. The value is stored in different memory location.
- In Java, all the method invocation are **call-by-value**.
- **Call-by-value** means that when you invoke a method, a copy of the value of each actual parameter is passed to the method. Any changes to the copy inside the method will have **no effect on the actual parameter**.



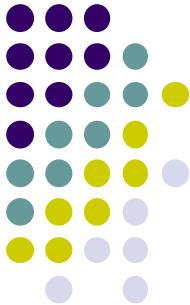
# Methods

- Object and array are known as **reference type**. Java does not manipulate objects and arrays directly. Instead, it manipulates **references to objects and arrays**.
- Thus, when object and array are used as parameter in the method, **any changes** to the instance variable of the object or items of the array **will have effect on the actual parameter**.



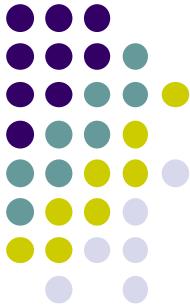
# Methods

- Each method contains two types of comment namely the **precondition** and the **postcondition**.
- Precondition states what is assumed to be true when the method is called.
- Postcondition describes the effect of the method call. It describes the value returned by the method.



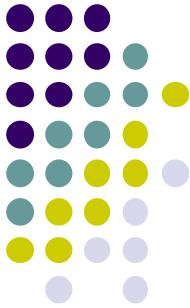
# Static Method

- Static method is the method that **do not require an calling object**.
- The definition of static method is inside the class definition.
- To define a static method
  - `public static returnType methodname(parameterType parameter, ...)`
- To invoke the a static method in the **same class**, just used the **methodname**.
- To invoke the a static method from **different class**, use **classname.methodname**.



# Static Method

- The main method is the static method.
  - `public static void main(String[] args)`
- The main method is used to start the program. It contains a parameter which is an array of String.



# Reference Type

- Reference types is the variable that contains references.
- In Java, a variable of class type stores the reference of where the object is located in the memory.
- Thus, a method can change the values of the instance variables of an argument of a class type.
- The Java primitive types are not the reference type.
- The constant **null** can be assigned to a variable of class type. null is used to indicate that the class type variable has no real value.

